```
serviceProvUserAdminAddressPkg PACKAGE
    BEHAVIOUR serviceProvUserAdminAddressPkgBehavior;
    ATTRIBUTES
        serviceProvUserAdminAddress GET-REPLACE;
    REGISTERED AS {LNP-OIDS.lnp-package 10};

serviceProvUserAdminAddressPkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        serviceProvUserAdminAddress attribute.
    !;

-- 11.0 LNP Service Provider Web Address Package

serviceProvWebAddressPkg PACKAGE
    BEHAVIOUR serviceProvWebAddressPkgBehavior;
    ATTRIBUTES
        serviceProvWebAddress GET-REPLACE;
    REGISTERED AS {LNP-OIDS.lnp-package 11};

serviceProvWebAddressPkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        serviceProvWebAddress attribute.
    !;

-- 12.0 LNP Subscription Version Activate Package

subscriptionVersionActivatePkg PACKAGE
    BEHAVIOUR subscriptionVersionActivatePkgBehavior;
    ACTIONS
        subscriptionVersionActivate;
    REGISTERED AS {LNP-OIDS.lnp-package 12};

subscriptionVersionActivatePkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionVersionActivate action.
    !;

-- 13.0 LNP Subscription Version Attribute Value Change Failed Service
--      Providers List

subscriptionVersionAttributeValueChangeFailed-SP-ListPkg PACKAGE
    BEHAVIOUR subscriptionVersionAttributeValueChangeFailed-SP-ListPkgBehavior;
    ATTRIBUTES
        subscriptionFailed-SP-List GET;
    REGISTERED AS {LNP-OIDS.lnp-package 13};

subscriptionVersionAttributeValueChangeFailed-SP-ListPkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionVersionAttributeValueChangeFailed-SP-List
        attribute.
    !;

-- 14.0 LNP Subscription Version Cancel Package

subscriptionVersionCancelPkg PACKAGE
    BEHAVIOUR subscriptionVersionCancelPkgBehavior;
    ACTIONS
        subscriptionVersionCancel;
    REGISTERED AS {LNP-OIDS.lnp-package 14};

subscriptionVersionCancelPkgBehavior BEHAVIOUR
    DEFINED AS !
```

```
        This package provides for conditionally including the
        subscriptionVersionCancel action.
    !;

-- 15.0 LNP Subscription Version Disconnect Package

subscriptionVersionDisconnectPkg PACKAGE
    BEHAVIOUR subscriptionVersionDisconnectPkgBehavior;
    ACTIONS
        subscriptionVersionDisconnect;
    REGISTERED AS {LNP-OIDS.lnp-package 15};

subscriptionVersionDisconnectPkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionVersionDisconnect action.
    !;

-- 16.0 LNP Subscription Version Local SMS Create Package

subscriptionVersionLocalSMS-CreatePkg PACKAGE
    BEHAVIOUR subscriptionVersionLocalSMS-CreatePkgBehavior;
    ACTIONS
        subscriptionVersionLocalSMS-Create;
    REGISTERED AS {LNP-OIDS.lnp-package 16};

subscriptionVersionLocalSMS-CreatePkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for including the
        subscriptionVersionLocalSMS-Create action.
    !;

-- 17.0 LNP Subscription Version Modify Package

subscriptionVersionModifyPkg PACKAGE
    BEHAVIOUR subscriptionVersionModifyPkgBehavior;
    ACTIONS
        subscriptionVersionModify;
    REGISTERED AS {LNP-OIDS.lnp-package 17};

subscriptionVersionModifyPkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionVersionModify action.
    !;

-- 18.0 LNP New Service Provider Subscription Version Cancellation
-- Acknowledge Package

subscriptionVersionNewSP-CancellationPkg PACKAGE
    BEHAVIOUR subscriptionVersionNewSP-CancellationPkgBehavior;
    ACTIONS
        subscriptionVersionNewSP-CancellationAcknowledge;
    REGISTERED AS {LNP-OIDS.lnp-package 18};

subscriptionVersionNewSP-CancellationPkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionVersionNewSP-CancellationAcknowledge action.
    !;

-- 20.0 LNP Subscription Version Remove From Conflict
-- Pending Package

subscriptionVersionRemoveFromConflictPkg PACKAGE
    BEHAVIOUR subscriptionVersionRemoveFromConflictPkgBehavior;
    ACTIONS
```

```
        subscriptionVersionRemoveFromConflict;
    REGISTERED AS {LNP-OIDS.lnp-package 20};

subscriptionVersionRemoveFromConflictPkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionVersionRemoveFromConflict action.
    !;

-- 21.0 LNP New Service Provider Subscription Version Create Package

subscriptionVersionNewSP-CreatePkg PACKAGE
    BEHAVIOUR subscriptionVersionNewSP-CreatePkgBehavior;
    ACTIONS
        subscriptionVersionNewSP-Create;
    REGISTERED AS {LNP-OIDS.lnp-package 21};

subscriptionVersionNewSP-CreatePkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionVersionNewSP-Create action.
    !;

-- 22.0 LNP Old Service Provider Subscription Version Cancellation
-- Acknowledge Package

subscriptionVersionOldSP-CancellationPkg PACKAGE
    BEHAVIOUR subscriptionVersionOldSP-CancellationPkgBehavior;
    ACTIONS
        subscriptionVersionOldSP-CancellationAcknowledge;
    REGISTERED AS {LNP-OIDS.lnp-package 22};

subscriptionVersionOldSP-CancellationPkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionVersionOldSP-CancellationAcknowledge action.
    !;

-- 24.0 LNP Old Service Provider Subscription Version Create Package

subscriptionVersionOldSP-CreatePkg PACKAGE
    BEHAVIOUR subscriptionVersionOldSP-CreatePkgBehavior;
    ACTIONS
        subscriptionVersionOldSP-Create;
    REGISTERED AS {LNP-OIDS.lnp-package 24};

subscriptionVersionOldSP-CreatePkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionVersionOldSP-Create action.
    !;

-- 25.0 LNP Subscription Status Change Cause Code Package

subscriptionStatusChangeCauseCodePkg PACKAGE
    BEHAVIOUR subscriptionStatusChangeCauseCodePkgBehavior;
    ATTRIBUTES
        subscriptionStatusChangeCauseCode GET;
    REGISTERED AS {LNP-OIDS.lnp-package 25};

subscriptionStatusChangeCauseCodePkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        subscriptionStatusChangeCauseCode attribute.
    !;

-- 26.0 LNP Service Provider Security Address Package
```

```
serviceProvSecurityAddressPkg PACKAGE
    BEHAVIOUR serviceProvSecurityAddressPkgBehavior;
    ATTRIBUTES
         serviceProvSecurityAddress GET-REPLACE;
    REGISTERED AS {LNP-OIDS.lnp-package 26};

serviceProvSecurityAddressPkgBehavior BEHAVIOUR
    DEFINED AS !
        This package provides for conditionally including the
        serviceProvSecurityAddress attribute.
    !;


--
```

7.6. -- Parameter Definitions

```
--

-- 1.0 Access Control Parameter

accessControlParameter PARAMETER
    CONTEXT EVENT-INFO;
    WITH SYNTAX LNP-ASN1.LnpAccessControl;
    REGISTERED AS {LNP-OIDS.lnp-parameter 1};

-- 2.0 LNP Specific Info Parameter

lnpSpecificInfoParameter PARAMETER
    CONTEXT SPECIFIC-ERROR;
--    ATTRIBUTE lnpSpecificInfo;
    WITH SYNTAX LNP-ASN1.LnpSpecificInfo;
    REGISTERED AS {LNP-OIDS.lnp-parameter 2};

--
```

7.7. -- Action Definitions

```
--

-- 1.0 LNP Download Action

lnpDownload ACTION
    BEHAVIOUR
        lnpDownloadDefinition,
        lnpDownloadBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.DownloadAction;
    WITH REPLY SYNTAX LNP-ASN1.DownloadReply;
    REGISTERED AS {LNP-OIDS.lnp-action 1};

lnpDownloadDefinition BEHAVIOUR
    DEFINED AS !
        The lnpDownload action is the action that is used by the Local SMS
        to specify the objects to be downloaded from the NPAC SMS.
    !;

lnpDownloadBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action is issued from an lnpSubscriptions
        or an lnpNetwork object and all objects to be downloaded
        are specified in the action request.

        Postconditions: After this action has been executed by the Local
        SMS specifying which objects to download, the NPAC SMS will
        determine which objects satisfy the download request and return
        them in the download action reply.

        Data to be downloaded can be specified by a time range of last
        modification/creation or by other criteria.  Time range requests
        will be limited to a tunable range specified in the NPAC SMS.
        All data modified/created in the download time period, regardless
        of the amount of data, will be downloaded.  For download requests
        not specifying a time range, the amount of data downloaded will be
        limited to a tunable amount as specified in the NPAC SMS.

        Criteria for a subscription download is a time range or a TN or
        TN range.

        Criteria for a network data download is a time range, service
        provider id or all service providers, an npa-nxx range or all
        npa-nxx data, an LRN range or all LRN data, or all network data.

        If a download requests fails in the NPAC SMS, the failure reason
        will be returned in the reply.
    !;

-- 2.0 LNP Recovery Complete Action

lnpRecoveryComplete ACTION
    BEHAVIOUR
        lnpRecoveryCompleteDefinition,
        lnpRecoveryCompleteBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.RecoveryCompleteAction;
    WITH REPLY SYNTAX LNP-ASN1.RecoveryCompleteReply;
    REGISTERED AS {LNP-OIDS.lnp-action 2};

lnpRecoveryCompleteDefinition BEHAVIOUR
    DEFINED AS !
        The lnpRecoveryComplete action is used by the Local SMS to
        specify the system has recovered from downtime and the
        transactions performed since the association establishment can now be
        sent from the NPAC SMS.
    !;

lnpRecoveryCompleteBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action is issued from an lnpLocalSMS object that
        specified the recovery mode flag in the access control as true at
        association establishment.

        Postconditions: After this action has been executed by the Local
        SMS specifying recovery is complete, the NPAC SMS will
        forward those updates which took place for the network and subscription
        data since the association was established in the action reply.

        If a recovery complete request fails in the NPAC SMS the failure reason
        will be returned in the reply.
    !;

-- 3.0 LNP Subscription Version Activate Action

subscriptionVersionActivate ACTION
    BEHAVIOUR
        subscriptionVersionActivateDefinition,
        subscriptionVersionActivateBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.ActivateAction;
    WITH REPLY SYNTAX LNP-ASN1.ActivateReply;
    REGISTERED AS {LNP-OIDS.lnp-action 3};

subscriptionVersionActivateDefinition BEHAVIOUR
```

```
    DEFINED AS !
        The subscriptionVersionActivate action is the action that can be
        used by the SOA of the new service provider to activate a
        subscription version id, tn or a range of tns via the SOA to
        NPAC SMS interface.
    !;

subscriptionVersionActivateBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action is issued from an lnpSubscriptions
        object specifying the object or range of objects to be activated by
        either subscriptionVersionId, the subscriptionTN or range of TNs.

        Postconditions: The service provider has activated the subscription
        version.  An error will be returned to the service provider if
        there is no version that can be activated at this time or if the
        activation fails due to the service provider not being the
        new service provider for the subscription version.

        Only pending subscription versions can be activated.  Attempts to
        port subscription that have not been authorized by both service
        providers will fail.

    !;

-- 4.0 LNP Subscription Version Cancel Action

subscriptionVersionCancel ACTION
    BEHAVIOUR
        subscriptionVersionCancelDefinition,
        subscriptionVersionCancelBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.CancelAction;
    WITH REPLY SYNTAX LNP-ASN1.CancelReply;
    REGISTERED AS {LNP-OIDS.lnp-action 4};

subscriptionVersionCancelDefinition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionCancel action is the action that can be
        used by the SOA to cancel a subscription version via the SOA to
        NPAC SMS interface.
    !;

subscriptionVersionCancelBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action is issued from an lnpSubscriptions
        object specifying the object or objects to be canceled by either
        the subscriptionVersionId, the subscriptionTN or a range of TNs.

        Postconditions: The service provider has set the version status
        to cancel-pending in the subscription version.  An error will be
        returned to the service provider if there is no version that can be
        canceled (i.e. pending, conflict, or disconnect-pending) or if the
        cancellation fails due to authorization of the service provider.
    !;

-- 5.0 LNP Subscription Version Disconnect Action

subscriptionVersionDisconnect ACTION
    BEHAVIOUR
        subscriptionVersionDisconnectDefinition,
        subscriptionVersionDisconnectBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.DisconnectAction;
    WITH REPLY SYNTAX LNP-ASN1.DisconnectReply;
    REGISTERED AS {LNP-OIDS.lnp-action 5};
```

```
subscriptionVersionDisconnectDefinition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionDisconnect action is the action that is
        used by the SOA to disconnect a subscription version via the SOA to
        NPAC SMS interface.
    !;

subscriptionVersionDisconnectBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action is issued from an lnpSubscriptions
        object and specifies the object or objects to be disconnected by
        either stating the subscriptionVersionId, the subscriptionTN or a
        range of TNs.  In addition, the customer's disconnect date is
        specified. An optional effective release date can be specified for
        a time deferred disconnect.

        Postconditions: The current service provider can disconnect an active
        subscription version.  An error will be returned to the service
        provider if there is no active version. If there is a pending
        version and the current service provider has NOT authorized the
        pending subscription version, the disconnect would take place
        and the pending subscription version would go into conflict.
        If the current service provider has authorized the pending subscription
        version, the NPAC SMS will fail the action back to the service
        provider.

        If the version is active, no outstanding versions exist,
        and the time stamp for disconnect has not been reached, the
        subscription version will be modified with a version status of
        disconnect-pending and the subscriptionEffectiveReleaseDate
        set to the effective release date specified in the action.

        If the version is active, there are no outstanding versions, and
        the time stamp for effective release has not been specified, the
        subscription version will be updated with a version status of
        sending.

        When the new subscription version status is set to sending either
        immediately or at the time the date and time specified in the
        subscriptionEffectiveReleaseDate, the broadcast time stamp is
        set to the current time when the disconnect version sending starts
        to the Local SMSs via the NPAC SMS to Local SMS interface.

        Before the broadcast of deletes begins, the
        subscriptionVersionDonorSP-CustomerDisconnectDate notification
        is sent to the donor SOA informing the service provider of the
        actual customer disconnect date.

        If the delete requests are successful for all Local SMSs, the
        current active version will have its version status marked as old
        and the subscriptionDisconnectCompleteTimeStamp is set to the
        current system date and time.

        If a delete request fails for the disconnect subscription
        version after the retry periods have expired, the version
        status will be set to failed or partially failed based on if the
        create failed in all or some of the Local SMSs respectively.
        The current active version will remain active and an error will be
        returned for the action.
    !;

-- 6.0 LNP Subscription Version Local SMS Create Action

subscriptionVersionLocalSMS-Create ACTION
    BEHAVIOUR
        subscriptionVersionLocalSMS-CreateDefinition,
        subscriptionVersionLocalSMS-CreateBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.LocalSMS-CreateAction;
    WITH REPLY SYNTAX LNP-ASN1.LocalSMS-CreateReply;
    REGISTERED AS {LNP-OIDS.lnp-action 6};

subscriptionVersionLocalSMS-CreateDefinition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionLocalSMS-Create action is the action that can be
        used by the NPAC SMS to create multiple subscription versions via the
        Local SMS to NPAC SMS interface.
    !;

subscriptionVersionLocalSMS-CreateBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action is issued from an lnpSubscriptions
        object specifying the object(s) to be created by
        the subscriptionVersionId and the subscriptionTN.  All attribute
        values required for creation will be supplied.

        Postconditions: A successful reply indicates the Local SMS can
        decipher the subscription version create action. An error will be
        returned to the NPAC SMS if the Local SMS cannot recognize the
        action data.

        The Local SMS will attempt to create all the specified subscription
        versions. It will return the subscriptionVersionActionResults
        notification to the NPAC SMS informing it of the success or
        failure of the creation attempts.
        !;

-- 7.0 LNP Subscription Version Modify Action

subscriptionVersionModify ACTION
    BEHAVIOUR
        subscriptionVersionModifyDefinition,
        subscriptionVersionModifyBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.ModifyAction;
    WITH REPLY SYNTAX LNP-ASN1.ModifyReply;
    REGISTERED AS {LNP-OIDS.lnp-action 7};

subscriptionVersionModifyDefinition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionModify action is the action that can be
        used by the SOA to modify a subscription version via the SOA to
        NPAC SMS interface.
    !;

subscriptionVersionModifyBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action is issued from an lnpSubscriptions
        object specifying the object to be modified by either
        the subscriptionVersionId, the subscriptionTN or a range of TNs and
        optionally the status of the subscription version.  All attribute
        values to be modified shall also be specified.

        Postconditions: The service provider has modified the subscription
        version.  An error will be returned to the service provider if
        there is no version that is modifiable or if the modification fails
        due to authorization of the service provider or data validation.

        Service Providers can modify attributes associated with active,
        pending, cancel-pending, disconnect-pending or conflict subscription
        versions.

        Old service providers can only modify the following attributes
        for pending, cancel-pending, or conflict subscription versions:
```

```
        subscriptionOldSP-DueDate
        subscriptionOldSP-Authorization
        subscriptionStatusChangeCauseCode

        The subscriptionStatusChangeCauseCode is an optional field and is
        only specified if the subscriptionOldSP-Authorization is false.

        New service providers can only modify the following attributes
        for pending, cancel-pending, or conflict subscription versions:

        subscriptionLRN
        subscriptionNewSP-DueDate
        subscriptionCLASS-DPC
        subscriptionCLASS-SSN
        subscriptionLIDB-DPC
        subscriptionLIDB-SSN
        subscriptionCNAM-DPC
        subscriptionCNAM-SSN
        subscriptionISVM-DPC
        subscriptionISVM-SSN
        subscriptionEndUserLocationValue
        subscriptionEndUserLocationType
        subscriptionBillingId

        Validation will be done for both old and new service provider
        data that is specified for pending, cancel-pending, or conflict
        subscription versions.
        If validation fails no changes will be made and an error
        will be returned. If the version passes validation, the version
        status will be set to pending if the subscriptionVersionStatus
        was not the attribute modified.  A new service provider can modify the
        subscriptionVersionStatus for a pending or disconnect-pending
        subscription version to cancel-pending.  An error
        message will be returned to the service provider if the status
        is not pending when they attempt to change the version
        status to cancel-pending.

        New service providers can only modify the following attributes
        for active subscription versions:

        subscriptionLRN
        subscriptionCLASS-DPC
        subscriptionCLASS-SSN
        subscriptionLIDB-DPC
        subscriptionLIDB-SSN
        subscriptionCNAM-DPC
        subscriptionCNAM-SSN
        subscriptionISVM-DPC
        subscriptionISVM-SSN
        subscriptionEndUserLocationValue
        subscriptionEndUserLocationType
        subscriptionBillingId

        If the data specified passes validation, the modified version
        is immediately activated.  The modified subscription version will have
        a status of sending and broadcasts will begin.  If validation fails,
        no changes will be made and an error will be returned in the action
        reply.

    !;

-- 8.0 LNP New Service Provider Cancellation Acknowledge Request

subscriptionVersionNewSP-CancellationAcknowledge ACTION
    BEHAVIOUR
        subscriptionVersionNewSP-CancellationAcknowledgeDefinition,
        subscriptionVersionNewSP-CancellationAcknowledgeBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.CancellationAcknowledgeAction;
    WITH REPLY SYNTAX LNP-ASN1.CancellationAcknowledgeReply;
    REGISTERED AS {LNP-OIDS.lnp-action 8};

subscriptionVersionNewSP-CancellationAcknowledgeDefinition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionNewSP-CancellationAcknowledge action
        is the action that is used the on NPAC SMS via the SOA to NPAC
        SMS interface by the new service provider to acknowledge
        cancellation of a subscriptionVersionNPAC with a status of
        cancel-pending.
    !;

subscriptionVersionNewSP-CancellationAcknowledgeBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action was issued from an lnpSubscriptions
        object specifying the object or objects to be acknowledged by either
        the subscriptionVersionId, the subscriptionTN or a range of
        subscriptionTNs.

        Postconditions: The service provider has acknowledged the
        subscription version.  An error will be returned to the service
        provider if no version exists that can have the cancellation
        acknowledged or if the acknowledgement fails due to
        the service provider not being authorized to perform the action.

        The subscriptionNewSP-CancellationTimeStamp will be
        updated to the current time if the action is successful and the
        version status will be changed to cancel if the old service
        provider has previously acknowledged the cancel.
    !;

-- 10.0 LNP Subscription Version Remove From Conflict

subscriptionVersionRemoveFromConflict ACTION
    BEHAVIOUR
        subscriptionVersionRemoveFromConflictDefinition,
        subscriptionVersionRemoveFromConflictBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.RemoveFromConflictAction;
    WITH REPLY SYNTAX LNP-ASN1.RemoveFromConflictReply;
    REGISTERED AS {LNP-OIDS.lnp-action 10};

subscriptionVersionRemoveFromConflictDefinition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionRemoveFromConflict action
        is the action that is used the on NPAC SMS via the SOA to NPAC
        SMS interface by either the old or new service provider to set the
        subscription version status from conflict to pending.
    !;

subscriptionVersionRemoveFromConflictBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action was issued from an lnpSubscriptions
        object specifying the object or objects to be updated by either
        the subscriptionVersionId, the subscriptionTN or a range of
        subscriptionTNs.

        Postconditions: The NPAC SMS has acknowledged the
        subscription version.  An error will be returned to the service
        provider if there is no version that can have the conflict
        status removed or if the service provider is not authorized to
        perform the action.

        Either the subscriptionNewSP-ConflictResolutionTimeStamp or
        subscriptionOldSP-ConflictResolutionTimeStamp will be updated to
        the current time if the action is successful and the
        version status will be changed from conflict to pending.

        If the old service provider issues the action, the
        subscriptionOldSP-AuthorizationTimeStamp is also updated to the
        current date and time.
    !;

-- 11.0 LNP New Service Provider Subscription Version Create

subscriptionVersionNewSP-Create ACTION
    BEHAVIOUR
        subscriptionVersionNewSP-CreateDefinition,
        subscriptionVersionNewSP-CreateBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.NewSP-CreateAction;
    WITH REPLY SYNTAX LNP-ASN1.NewSP-CreateReply;
    REGISTERED AS {LNP-OIDS.lnp-action 11};

subscriptionVersionNewSP-CreateDefinition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionNewSP-Create action is the action that is
        used the on NPAC SMS via the SOA to NPAC SMS interface by the
        new service provider to create a new subscriptionVersionNPAC.
    !;

subscriptionVersionNewSP-CreateBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action is issued from an lnpSubscriptions
        object.  Creates can only be performed provided there is only one
        currently active subscription or an action failure will be returned.

        The new service provider must specify valid values for the
        following attributes:

        subscriptionTN or a valid subscriptionVersionTN-Range
        subscriptionLRN
        subscriptionNewCurrentSP
        subscriptionOldSP
        subscriptionNewSP-DueDate
        subscriptionCLASS-DPC
        subscriptionCLASS-SSN
        subscriptionLIDB-DPC
        subscriptionLIDB-SSN
        subscriptionCNAM-DPC
        subscriptionCNAM-SSN
        subscriptionISVM-DPC
        subscriptionISVM-SSN
        subscriptionLNPType
        subscriptionPortingToOriginal-SPSwitch

        The new service provider may specify valid values for the
        following attributes:

        subscriptionEndUserLocationValue
        subscriptionEndUserLocationType
        subscriptionBillingId

        subscriptionPortingToOriginal-SPSwitch can only be specified as
        TRUE for a TN that is currently ported and is being ported back
        to the original service provider.  If the value of
        subscriptionPortingToOriginal-SPSwitch is TRUE, the LRN and GTT data
        should be specified as NULL.  If the variable is TRUE,
        when the activate occurs for the subscription version, the Local
        SMS's will receive a request to delete the old subscription version
        routing data in their networks. They will not receive any
        new network routing data for the subscription. Concurrence from the
        old service provider is required.

        If the port of the subscription version is an intra-service
        provider port, the new service provider can use the
        subscriptionVersionNewSP-Create action specifying the old service
        provider equal to the new service provider.  In this case, the
        old service provider create action is not required.

        Postconditions: After this action has been executed, if
        the data specified passes validation, a pending subscription
        version will exist in the NPAC SMS.  These validations are
        done as follows:

        subscriptionTN or range of TNs are valid in a range open for
        porting by the old service provider.

        subscriptionLNPType is specified to be "LSPP" or "LISP".

        subscriptionNewSP-DueDate is a future date.  If not specified,
        the time defaults to 00:00.00.

        Old and New SP are valid service providers in the NPAC SMS.

        LRN data is associated with the New Service Provider.

        If a pre-existing version exists, validation will be done to insure
        that the new service provider previously specified is the same
        as the executor of the action.

        If the validations succeed and the subscription version does not exist,
        a new subscription version will be created with a status of pending.

        If the validations succeed and the subscription version already
        exists, the new service provider data will be applied to the
        subscription version.

        If the validations fail, a new subscription version will not
        be created if one does not exist.  If one already existed, it
        will be retained.

        The action success or failure and reasons for failure will be
        returned in the action reply.
    !;

-- 12.0 LNP Old Service Provider Cancellation Acknowledge Request

subscriptionVersionOldSP-CancellationAcknowledge ACTION
    BEHAVIOUR
        subscriptionVersionOldSP-CancellationAcknowledgeDefinition,
        subscriptionVersionOldSP-CancellationAcknowledgeBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.CancellationAcknowledgeAction;
    WITH REPLY SYNTAX LNP-ASN1.CancellationAcknowledgeReply;
    REGISTERED AS {LNP-OIDS.lnp-action 12};

subscriptionVersionOldSP-CancellationAcknowledgeDefinition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionOldSP-CancellationAcknowledge action
        is the action that is used on the NPAC SMS via the SOA to NPAC
        SMS interface by the old service provider to acknowledge
        cancellation of a subscriptionVersionNPAC with a status of
        cancel-pending.
    !;

subscriptionVersionOldSP-CancellationAcknowledgeBehavior BEHAVIOUR
    DEFINED AS !
```

```
        Preconditions: This action was issued from an lnpSubscriptions
        object specifying the object or objects to be acknowledged by either
        the subscriptionVersionId, the subscriptionTN or a range of
        subscriptionTNs.

        Postconditions: The service provider has acknowledged the
        subscription version.  An error will be returned to the service
        provider if there is no version that can have cancellation
        acknowledged or if the acknowledgement fails due to
        the service provider not being authorized to perform the action.

        The subscriptionOldSP-CancellationTimeStamp will be
        updated to the current time if the action is successful and the
        version status will be changed to cancel if the new service
        provider has previously acknowledged the cancel.
    !;

-- 14.0 LNP Old Service Provider Subscription Version Create

subscriptionVersionOldSP-Create ACTION
    BEHAVIOUR
        subscriptionVersionOldSP-CreateDefinition,
        subscriptionVersionOldSP-CreateBehavior;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX LNP-ASN1.OldSP-CreateAction;
    WITH REPLY SYNTAX LNP-ASN1.OldSP-CreateReply;
    REGISTERED AS {LNP-OIDS.lnp-action 14};

subscriptionVersionOldSP-CreateDefinition BEHAVIOUR
    DEFINED AS !
        The subscriptionVersionOldSP-Create action is the action that is
        used the on NPAC SMS via the SOA to NPAC SMS interface by the
        old service provider to create a new subscriptionVersionNPAC.
    !;

subscriptionVersionOldSP-CreateBehavior BEHAVIOUR
    DEFINED AS !
        Preconditions: This action was issued from an lnpSubscriptions
        object.  Creates can be performed provided there is only one
        currently active subscription or action failure will be returned.

        The old service provider must specify valid values for the
        following attributes:

        subscriptionTN or a valid subscriptionVersionTN-Range
        subscriptionNewCurrentSP
        subscriptionOldSP
        subscriptionOldSP-DueDate
        subscriptionOldSP-Authorization
        subscriptionLNPType

        If the subscriptionOldSP-Authorization is false, the old service
        provider must specify a subscriptionStatusChangeCauseCode.

        Postconditions: After this action has been executed if
        the data specified passes validation, a pending subscription
        version will exist in the NPAC SMS.  These validations are
        done as follows:

        subscriptionTN or range of TNs are valid in a range open for
        porting by the old service provider.

        subscriptionLNPType is specified as "LSPP" or "LISP".

        subscriptionOldSP-DueDate is a future date. If not specified,
        the time defaults to 00:00.00.
        Old and New SP are valid service providers in the NPAC SMS and
        the new service provider is not equal to the old service provider.

        If a pre-existing version exists, validation will be done to insure
        that the new service provider previously specified is the same
        as the executor of the action.

        If the validations succeed and the subscription version does not exist,
        a new subscription version will be created with a status of pending.

        If the validations succeed and the subscription version already
        exists, the old service provider data will be applied to the
        existing subscription version.

        If the validations fail, a new subscription version will not
        be created if one does not exist.  If one already existed it
        will be retained and an error returned.

        The action success or failure and reasons for failure will be
        returned in the action reply.
    !;

--
```

7.8. -- Notification Definitions

```
--

-- 1.0 LNP NPAC SMS Operational Information Notification

lnpNPAC-SMS-Operational-Information NOTIFICATION
    BEHAVIOUR  lnpNPAC-SMS-Operational-InformationBehavior;
    WITH INFORMATION SYNTAX LNP-ASN1.NPAC-SMS-Operational-Information
    AND ATTRIBUTE IDS
        down-time downTime,
        npac-contact-number npacContactNumber,
        additional-down-time-information additionalDownTimeInformation,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 1};

lnpNPAC-SMS-Operational-InformationBehavior BEHAVIOUR
    DEFINED AS !
        This notification contains information about the NPAC SMS's
        scheduled down time.  This notification contains the start and
        stop date and time for the planned down time. It is sent to both the
        SOA and Local SMS systems.
    !;

-- 2.0 LNP Subscription Audit Local SMS Discrepancy Report

subscriptionAudit-DiscrepancyRpt NOTIFICATION
    BEHAVIOUR  subscriptionAudit-DiscrepancyRptBehavior;
    WITH INFORMATION SYNTAX LNP-ASN1.AuditDiscrepancyRpt
    AND ATTRIBUTE IDS
        tn auditDiscrepancyTn,
        version-id auditDiscrepancyVersionId,
        lsms-service-prov-id auditDiscrepancyLSMS-SP-Id,
        failure-reason auditDiscrepancyFailureReason,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 2};

subscriptionAudit-DiscrepancyRptBehavior BEHAVIOUR
    DEFINED AS !
        This notification contains a report on a discrepancy found during
        an audit.  The discrepancy contains the subscription TN and Version
        ID for which the discrepancy was found and the error.  Valid
        errors are:
```

```
        fields mismatched between NPAC SMS and Local SMS
        record missing in Local SMS and associated Service Provider Id
        record missing in NPAC SMS

        If field mismatches are found then the attribute(s) for which the
        mismatch and the Local SMS value(s) and the NPAC SMS value(s)
        will be returned as well as the Service Provider Id associated
        with the Local SMS.

        When audit discrepancy notifications are sent to the NPAC SMS by
        the Local SMS create or modification requests to correct the
        discrepancy will be done by the NPAC SMS.
    !;

-- 3.0 LNP Subscription Audit Results

subscriptionAuditResults NOTIFICATION
    BEHAVIOUR  subscriptionAuditResultsBehavior;
    WITH INFORMATION SYNTAX LNP-ASN1.AuditResults
    AND ATTRIBUTE IDS
        status auditResultStatus,
        failed-service-prov-list auditResultFailed-SP-List,
        number-of-discrepancies auditResultNumberDiscrepancies,
        time-of-completion auditResultCompletionTime,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 3};

subscriptionAuditResultsBehavior BEHAVIOUR
    DEFINED AS !
        This notification contains the results of an audit.  It contains
        the name of the audit, the number of discrepancies found during the
        audit, the success or failure of the audit, and the time of audit
        completion or failure.
    !;

-- 4.0 LNP Subscription Version Cancellation Resolution Request
-- Notification

subscriptionVersionCancellationAcknowledgeRequest NOTIFICATION
    BEHAVIOUR  subscriptionVersionCancellationAcknowledgeBehavior;
    WITH INFORMATION SYNTAX
        LNP-ASN1.VersionCancellationAcknowledgeRequest
    AND ATTRIBUTE IDS
        tn subscriptionTN,
        version-id subscriptionVersionId,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 4};

subscriptionVersionCancellationAcknowledgeBehavior BEHAVIOUR
    DEFINED AS !
        This notification requests that a service provider send
        a cancellation acknowledgement for a subscription
        version.  The TN and the version id are sent.
    !;

-- 6.0 LNP Subscription Version Donor Service Provider Customer Disconnect Date
--     Notification

subscriptionVersionDonorSP-CustomerDisconnectDate NOTIFICATION
    BEHAVIOUR  subscriptionVersionDonorSP-CustomerDisconnectDateBehavior;
    WITH INFORMATION SYNTAX LNP-ASN1.VersionCustomerDisconnectDate
    AND ATTRIBUTE IDS
        tn subscriptionTN,
        version-id subscriptionVersionId,
        service-prov-customer-disconnect-date
            subscriptionCustomerDisconnectDate,
        service-prov-effective-release-date
            subscriptionEffectiveReleaseDate,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 6};

subscriptionVersionDonorSP-CustomerDisconnectDateBehavior BEHAVIOUR
    DEFINED AS !
        This notification informs the donor service provider SOA
        that a subscription version is being disconnected.
        The TN, the version id, customer disconnect date and
        effective release date values are sent.
    !;


-- 7.0 LNP Subscription Version Local SMS Action Results

subscriptionVersionLocalSMS-ActionResults NOTIFICATION
    BEHAVIOUR  subscriptionVersionLocalSMS-ActionResultsBehavior;
    WITH INFORMATION SYNTAX LNP-ASN1.LocalSMS-ActionResults
    AND ATTRIBUTE IDS
        actionId actionId,
        status actionResultsStatus,
        failed-tn-list failedTN-List,
        time-of-completion resultsCompletionTime,
        accessControl accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 7};

subscriptionVersionLocalSMS-ActionResultsBehavior BEHAVIOUR
    DEFINED AS !
        This notification contains the reuslts of a
        subscriptionVersionLocalSMS-Create action once all the create requests
        have been attempted. It contains the id of the action, the success or
        failure of the action, the completion time and an optional list of
        failed subscriptionTNs and error codes.
    !;

-- 8.0 LNP Subscription Version New NPA-NXX Notification

subscriptionVersionNewNPA-NXX NOTIFICATION
    BEHAVIOUR  subscriptionVersionNewNPA-NXXBehavior;
    WITH INFORMATION SYNTAX
        LNP-ASN1.VersionNewNPA-NXX
    AND ATTRIBUTE IDS
        service-prov-npa-nxx-id serviceProvNPA-NXX-ID,
        service-prov-npa-nxx-value serviceProvNPA-NXX-Value,
        service-prov-npa-nxx-effective-time-stamp
          serviceProvNPA-NXX-EffectiveTimeStamp,
        service-prov-id serviceProvID,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 8};

subscriptionVersionNewNPA-NXXBehavior BEHAVIOUR
    DEFINED AS !
        This notification informs the SOA and Local SMS of a pending
        subscription version involving a new NPA-NXX.  The
        service-prov-npa-nxx-id, service-prov-npa-nxx-value,
        service-prov-npa-nxx-effective-time-stamp and service-prov-id
        are sent.
    !;

-- 9.0 LNP Subscription Version New SP Create Request Notification

subscriptionVersionNewSP-CreateRequest NOTIFICATION
    BEHAVIOUR  subscriptionVersionNewSP-CreateRequestBehavior;
    WITH INFORMATION SYNTAX LNP-ASN1.VersionNewSP-CreateRequest
    AND ATTRIBUTE IDS
        tn subscriptionTN,
        version-id subscriptionVersionId,
        service-prov-id subscriptionOldSP,
        service-prov-due-date subscriptionOldSP-DueDate,
        service-prov-authorization subscriptionOldSP-Authorization,
        service-prov-authorization-time-stamp
            subscriptionOldSP-AuthorizationTimeStamp,
        status-change-cause-code subscriptionStatusChangeCauseCode,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 9};

subscriptionVersionNewSP-CreateRequestBehavior BEHAVIOUR
    DEFINED AS !
        This notification requests that a new service provider send
        a create request for a subscription version for which
        concurrence for porting the number has not been received.
        The TN, the version id and the old service provider id,
        authorization flag and authorization timestamp values are sent.
    !;


-- 10.0 LNP Subscription Version Old SP Concurrence Request Notification

subscriptionVersionOldSP-ConcurrenceRequest NOTIFICATION
    BEHAVIOUR  subscriptionVersionOldSP-ConcurrenceRequestBehavior;
    WITH INFORMATION SYNTAX LNP-ASN1.VersionOldSP-ConcurrenceRequest
    AND ATTRIBUTE IDS
        tn subscriptionTN,
        version-id subscriptionVersionId,
        service-prov-id subscriptionNewCurrentSP,
        service-prov-due-date subscriptionNewSP-DueDate,
        service-prov-creation-time-stamp
            subscriptionNewSP-CreationTimeStamp,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 10};

subscriptionVersionOldSP-ConcurrenceRequestBehavior BEHAVIOUR
    DEFINED AS !
        This notification requests that a old service provider send
        a create request for a subscription version for which
        concurrence for porting the number has not been received.
        The TN, the version id, and the new service provider id,
        authorization flag and creation timestamp values are sent.
    !;

-- 11.0 LNP Subscription Version Status Attribute Value Change Notification

subscriptionVersionStatusAttributeValueChange NOTIFICATION
    BEHAVIOUR  subscriptionVersionStatusAttributeValueChangeBehavior;
    WITH INFORMATION SYNTAX  LNP-ASN1.VersionStatusAttributeValueChange
    AND ATTRIBUTE IDS
        value-change-info subscriptionVersionAttributeValueChangeInfo,
        failed-service-provs subscriptionFailed-SP-List,
        status-change-cause-code subscriptionStatusChangeCauseCode,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 11};

subscriptionVersionStatusAttributeValueChangeBehavior BEHAVIOUR
    DEFINED  AS !
        This notification type is used to report changes to the
        subscriptionVersionStatus field.  It is identical to an
        attribute value change notification as defined in M.3100
        except for the addition of the list of failed service
        providers in cases where the version status is active, failed or
        partially failed and the subscriptionStatusChangeCauseCode if
        it is set.

        Failed lists will also be potentially sent for subscription versions
        with statuses of disconnect-pending and old.
    !;

-- 12.0 LNP Subscription Version Final Concurrence Timer Expiration
--      Notification

subscriptionVersionOldSPFinalConcurrenceWindowExpiration NOTIFICATION
    BEHAVIOUR  subscriptionVersionOldSPFinalConcurrenceWindowExpirationBehavior;
    WITH INFORMATION SYNTAX
        LNP-ASN1.VersionOldSPFinalConcurrenceWindowExpiration
    AND ATTRIBUTE IDS
        tn subscriptionTN,
        version-id subscriptionVersionId,
        access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 12};

subscriptionVersionOldSPFinalConcurrenceWindowExpirationBehavior BEHAVIOR
    DEFINED AS !
        This notification will be sent by the NPAC SMS upon expiration of
        the Final Concurrence Timer to the old service provider via the SOA
        to NPAC SMS interface to inform them of the timer expiration.
    !;
```

8. General ASN.1 Definitions

8

8.1. Overview

The ASN.1 definitions provided below support the GDMO definitions in Chapter 5. Included below are the ASN.1 object identifier definitions and the syntax definitions for the interface attributes, notifications, and actions.

8.2. LNP ASN.1 Object Identifier Definitions

```
--#include    "smi.asn"

LNP-OIDS
  {joint-iso-ccitt(2) country(16) us(840) organization(1)
   lnp-net(4) oids(0)}

DEFINITIONS ::=

BEGIN

-- EXPORTS all definitions

lnp-net OBJECT IDENTIFIER ::=
  {joint-iso-ccitt(2) country(16) us(840) organization(1)
   lnp-net(4) }

-- LNP categories of interface information objects

lnp-attribute OBJECT IDENTIFIER ::= {lnp-net attribute(1) }
lnp-objectClass OBJECT IDENTIFIER ::= {lnp-net objectClass(2) }
lnp-nameBinding OBJECT IDENTIFIER ::= {lnp-net nameBinding(3) }
lnp-notification OBJECT IDENTIFIER ::= {lnp-net notification(4) }
lnp-action OBJECT IDENTIFIER ::= {lnp-net action(5) }
lnp-package OBJECT IDENTIFIER ::= {lnp-net package(6) }
lnp-parameter OBJECT IDENTIFIER ::= {lnp-net parameter(7) }

END -- LNP-OIDS
```

8.3. LNP General ASN.1 Definitions

```
LNP-ASN1
  {joint-iso-ccitt(2) country(16) us(840) organization(1)
   lnp-net(4) asn1(1)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS everything

IMPORTS
```

```
-- CMIP
 ObjectClass, ObjectInstance
        FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)}

-- DMI

 AttributeValueChangeInfo
        FROM Notification-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2)
             asn1Module(2) 2};

ActivateAction ::= SubscriptionVersionAction

ActionResultsStatus ::= ResultsStatus

ActivateReply ::= SubscriptionVersionActionReply

AddressInformation ::= SEQUENCE {
    line1 GraphicString40,
    line2  GraphicString40,
    city   GraphicString20,
    state  GraphicString(SIZE(2)),
    zip  GraphicString(SIZE(9)),
    province GraphicString(SIZE(2)),
    country GraphicString20,
    contactPhone  PhoneNumber,
    contact  GraphicString40,
    contactFax  PhoneNumber,
    contactPager  PhoneNumber,
    contactPagerPIN  DigitString,
    contactE-mail  GraphicString60
}

AssociationFunction ::= SEQUENCE {
    soaUnits SoaUnits,
    lsmsUnits LSMSUnits
}

AuditAttributes ::= CHOICE {
    specific-audit [0] SEQUENCE {
        lidb-data BOOLEAN,
        class-data BOOLEAN,
        cnam-data BOOLEAN,
        isvm-data BOOLEAN,
        lrn-data BOOLEAN
    },
    all-data [1] NULL
}

AuditDiscrepancyRpt ::= SEQUENCE {
    tn PhoneNumber,
    version-id SubscriptionVersionId,
    lsms-service-prov-id ServiceProvId,
    failure-reason AuditFailureData,
    access-control LnpAccessControl
}

AuditFailureData ::= CHOICE {
    tn-version-missing-NPAC [0] NULL,
    tn-version-missing-LSMS [1] NULL,
    mismatch-data [2] MismatchAttributes
}

AuditId ::= LnpKey

AuditName ::= GraphicString40

AuditNumberOfTNs ::= INTEGER
```

```
AuditNumberOfTNsComplete ::= INTEGER

AuditResults ::= SEQUENCE {
    status [0] AuditResultStatus,
    failed-service-prov-list [1] Failed-SP-List OPTIONAL,
    number-of-discrepancies [2] INTEGER,
    time-of-completion [3] GeneralizedTime,
    access-control [4] LnpAccessControl
}

AuditResultStatus ::= ENUMERATED {
    success (0),
    failed-due-to-discrepancies (1),
    failed-on-local-sms (2),
    no-audit-performed (3)
}

AuditServiceProvIdRange ::= CHOICE {
    allServiceProvs [0] NULL,
    serviceProv [1] ServiceProvName
}

AuditStatus ::= ENUMERATED {
    in-progress (0),
    suspended (1),

   complete (2)
}

AuditTN-ActivationRange ::= TimeRange

BillingId ::= CHOICE {
    value [0] GraphicString4,
    no-value-needed [1] NULL
}

Boolean ::= BOOLEAN

CancellationAcknowledgeAction ::= SubscriptionVersionAction

CancellationAcknowledgeReply ::= SubscriptionVersionActionReply

CancelAction::= SubscriptionVersionAction

CancelReply ::= SubscriptionVersionActionReply

DPC ::= CHOICE {
    dpc-value       [0] OCTET STRING (SIZE(3)),
    no-value-needed [1] NULL
}

DigitString ::= GraphicString (FROM ("0" | "1" | "2" | "3" | "4" | "5" |
                "6" | "7" | "8" | "9" | "*" | "#" ))

DisconnectAction::= SEQUENCE {
    subscription-version-action [0] EXPLICIT SubscriptionVersionAction,
    customer-disconnect-date [1] GeneralizedTime,
    effective-release-date [2] GeneralizedTime OPTIONAL
}

DisconnectReply ::= SEQUENCE {
    status SubscriptionVersionActionReply,
    version-id SET OF SubscriptionVersionId OPTIONAL
}

DownloadAction ::= CHOICE {
    subscriber-download [0] EXPLICIT SubscriptionDownloadCriteria,
    network-download [1] NetworkDownloadCriteria
}

DownloadReason ::= ENUMERATED {
    new1 (0),
    delete1(1),
    modified (2),
    audit-discrepancy (3)
}

DownloadReply ::= SEQUENCE {
    status ENUMERATED {
        success (0),
        failed (1),
        time-range-invalid (2),
        criteria-to-large (3),
        no-data-selected (4)
    },
    downloaddata CHOICE {
        subscriber-data [0] SubscriptionDownloadData,
        network-data [1] NetworkDownloadData
    } OPTIONAL
}

EndUserLocationType ::= CHOICE {
    value [0] NumberString(SIZE(2)),
    no-value-needed [1] NULL
}

EndUserLocationValue ::= CHOICE {
    value [0] NumberString(SIZE(1..12)),
    no-value-needed [1] NULL
}

Failed-SP-List  ::= SET OF SEQUENCE {
    service-prov-id ServiceProvId,
    service-prov-name ServiceProvName
}

CMIPErrorCode ::= ENUMERATED {
    noSuchObjectClassEr (0),
    noSuchObjectInstanceEr (1),
    accessDeniedEr (2),
    syncNotSupportedEr (3),
    invalidFilterEr (4),
    noSuchAttributeEr (5),
    invalidAttributeValueEr (6),
    getListErrorEr (7),
    setListErorrEr (8),
    noSuchActionEr (9),
    processingFailureEr (10),
    duplicateManagedObjectInstanceEr (11),
    noSuchReferenceObjectEr (12),
    noSuchEventTypeEr (13),
    noSuchArgumentEr (14),
    invalidArgumentValueEr (15),
    invalidScopeEr (16),
    invalidObjectInstanceEr (17),
    missingattributeValueEr (18),
    classInstanceConflictEr (19),
    complexityLimitationEr (20),
    mistypedOperationEr (21),
    noSuchInvokeIdEr (22),
    operationCancelledEr (23)
}
```

```
FailedTN-List  ::= SET OF SEQUENCE {
    subscriptionVersionId SubscriptionVersionId,
    tn PhoneNumber,
    errorId CMIPErrorCode
}

GeneralTime ::= GeneralizedTime

GraphicStringBase ::= GraphicString

GraphicString4 ::= GraphicStringBase(SIZE(1..4))

GraphicString16 ::= GraphicStringBase(SIZE(1..16))

GraphicString20 ::= GraphicStringBase(SIZE(1..20))

GraphicString25 ::= GraphicStringBase(SIZE(1..25))

GraphicString28 ::= GraphicStringBase(SIZE(1..28))

GraphicString40 ::= GraphicStringBase(SIZE(1..40))

GraphicString60 ::= GraphicStringBase(SIZE(1..60))

GraphicString255 ::= GraphicStringBase(SIZE(1..255))

Integer ::= INTEGER

LnpAccessControl ::= [0] SEQUENCE {
    systemId [0] EXPLICIT SystemID,
    systemType [1] SystemType,
    userId [2] GraphicString60 OPTIONAL,
    listId [3] INTEGER,
    keyId [4] INTEGER,
    cmipDepartureTime [5] GeneralizedTime,
    sequenceNumber [6] INTEGER(0..4294967295),
    function [7] AssociationFunction,
    recoveryMode [8] BOOLEAN,
    signature [9] BIT STRING
}

LnpAuditsName ::= GraphicString ("lnpAudits")

LnpKey ::= INTEGER

LnpNetworkName ::= GraphicString ("lnpNetwork")

LnpSMS-Name ::= GraphicString40

LnpServiceProvsName ::= GraphicString ("lnpServiceProvs")

LnpSubscriptionsName ::= GraphicString ("lnpSubscriptions")

LnpSpecificInfo ::= GraphicString255


LNPType ::= ENUMERATED {
    lspp (0),
    lisp (1)
}

LocalSMS-ActionResults ::= SEQUENCE {
    actionId [0] INTEGER,
    status [1] ActionResultsStatus,
    failed-tn-list [2] FailedTN-List OPTIONAL,
    time-of-completion [3] GeneralizedTime,
    accessControl [4] LnpAccessControl
```

```
}

LocalSMS-CreateAction ::= SEQUENCE {
    actionId INTEGER,
    subscriptionVersionObjects SET OF SubscriptionVersionObject
}

LocalSMS-CreateReply ::= ResultsStatus

LRN ::= CHOICE {
    value [0] OCTET STRING (SIZE(5)),
    no-value-needed [1] NULL
}


LRN-ID ::= LnpKey

LRN-DownloadData ::= SET OF SEQUENCE {
    service-prov-lrn-id LRN-ID,
    service-prov-lrn-value LRN,
    service-prov-download-reason DownloadReason,
    service-prov-lrn-creation-timestamp GeneralizedTime OPTIONAL
}

LRN-Range ::= SEQUENCE {
  start-lrn LRN,
  stop-lrn LRN
}

LSMSUnits ::= SEQUENCE {
    dataDownload [0] NULL OPTIONAL,
    networkDataMgmt [1] NULL OPTIONAL,
    query [2] NULL OPTIONAL
}

MismatchAttributes ::= SEQUENCE {
    seq0 [0] SEQUENCE {
        lsms-subscriptionLRN LRN,
        npac-subscriptionLRN LRN
    } OPTIONAL,
    seq1 [1] SEQUENCE {
        lsms-subscriptionNewCurrentSP ServiceProvId,
        npac-subscriptionNewCurrentSP ServiceProvId
    } OPTIONAL,
    seq2 [2] SEQUENCE {
        lsms-subscriptionActivationTimeStamp GeneralizedTime,
        npac-subscriptionActivationTimeStamp GeneralizedTime
    } OPTIONAL,
    seq3 [3] SEQUENCE {
        lsms-subscriptionCLASS-DPC DPC,
        npac-subscriptionCLASS-DPC DPC
    } OPTIONAL,
    seq4 [4] SEQUENCE {
        lsms-subscriptionCLASS-SSN SSN,
        npac-subscriptionCLASS-SSN SSN
    } OPTIONAL,
    seq5 [5] SEQUENCE {
        lsms-subscriptionLIDB-DPC DPC,
        npac-subscriptionLIDB-DPC DPC
    } OPTIONAL,
    seq6 [6] SEQUENCE {
        lsms-subscriptionLIDB-SSN SSN,
        npac-subscriptionLIDB-SSN SSN
    } OPTIONAL,
    seq7 [7] SEQUENCE {
        lsms-subscriptionISVM-DPC DPC,
        npac-subscriptionISVM-DPC DPC
    } OPTIONAL,
    seq8 [8] SEQUENCE {
        lsms-subscriptionISVM-SSN SSN,
        npac-subscriptionISVM-SSN SSN
    } OPTIONAL,
    seq9 [9] SEQUENCE {
        lsms-subscriptionCNAM-DPC DPC,
        npac-subscriptionCNAM-DPC DPC
    } OPTIONAL,
    seq10 [10] SEQUENCE {
        lsms-subscriptionCNAM-SSN SSN,
        npac-subscriptionCNAM-SSN SSN
    } OPTIONAL,
    seq11 [11] SEQUENCE {
        lsms-subscriptionEndUserLocationValue EndUserLocationValue,
        npac-subscriptionEndUserLocationValue EndUserLocationValue
    } OPTIONAL,
    seq12 [12] SEQUENCE {
        lsms-subscriptionEndUserLocationType EndUserLocationType,
        npac-subscriptionEndUserLocationType EndUserLocationType
    } OPTIONAL,
    seq13 [13] SEQUENCE {
        lsms-subscriptionBillingId BillingId,
        npac-subscriptionBillingId BillingId
    } OPTIONAL,
    seq14 [14] SEQUENCE {
        lsms-subscriptionLNPType LNPType,
        npac-subscriptionLNPType LNPType
    } OPTIONAL
}

ModifyAction::= SEQUENCE {
    subscription-version-action [0] EXPLICIT SubscriptionVersionAction,
    version-status [1] VersionStatus OPTIONAL,
    data-to-modify [2] SubscriptionModifyData
}

ModifyReply ::= SEQUENCE {
    status SubscriptionVersionActionReply,
    invalid-data SubscriptionModifyInvalidData OPTIONAL
}

NetworkAddressInformation ::= SET OF SEQUENCE {
    interfaceAddress OSI-Address,
    systemType SystemType
}

NetworkDownloadCriteria ::= SEQUENCE {
    time-range [0] TimeRange OPTIONAL,
    chc1 [1] EXPLICIT CHOICE {
        service-prov [0] ServiceProvId,
        all-service-provs [1] NULL
    },
    chc2 [2] EXPLICIT CHOICE {-- A decision was made by
                              -- NANC to leave this structure a CHOICE of
                              -- CHOICEs instead of using one CHOICE to
                              -- simplify tagging
        npa-nxx-data [0] EXPLICIT CHOICE {
            npa-nxx-range [0] NPA-NXX-Range,

           all-npa-nxx [1] NULL
        },
        lrn-data [1] EXPLICIT CHOICE {
            lrn-range [0] LRN-Range,
            all-lrn [1] NULL
        },
        all-network-data [2] NULL
```